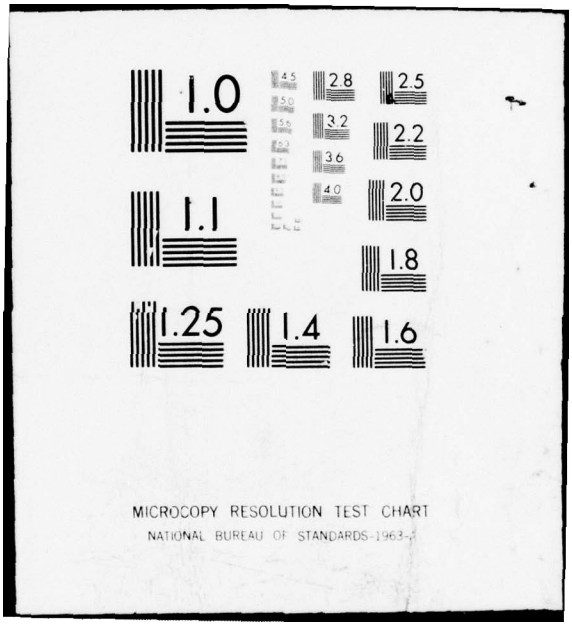
AD-A038 054 AIR FORCE AVIONICS LAB WRIGHT-PATTERSON AFB OHIO F/G 9/2
REPORT ON HIGH-ORDER LANGUAGE (HOL) STANDARDIZATION FOR AVIONIC--ETC(U)
JAN 77 W L TRAINOR

UNCLASSIFIED AFAL-TR-76-254 NL

OF
AD A038054

END

DATE FILMED
4-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFAL-TR-76-254

(12)

REPORT ON HIGH-ORDER LANGUAGE (HOL) STANDARDIZATION FOR AVIONICS

DAIS PROJECT OFFICE
SYSTEM AVIONICS DIVISION

JANUARY 1977

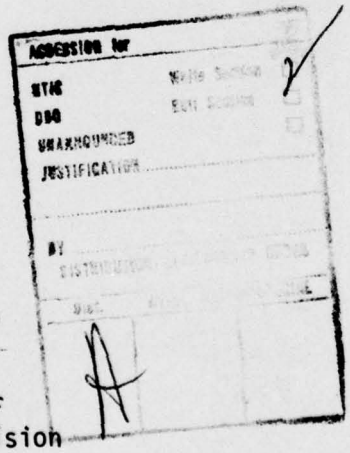
TECHNICAL REPORT AFAL-TR-76-254
FINAL REPORT FOR AUGUST - SEPTEMBER 1976

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto. This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NITS). At NITS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



W. LYNN TRAINOR
Technical Manager,
DAIS Software Group

FOR THE COMMANDER

JAMES D. EVERETT, Col, USAF
Chief, System Avionics Division
AF Avionics Laboratory

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 4 APRIL 77 - 100

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAL-TR-76-254	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Report on High-Order Language (HOL) Standardization for Avionics		5. TYPE OF REPORT & PERIOD COVERED Final Report Aug – Sep 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mr. W. Lynn Trainor		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Avionics Laboratory (AFAL/AA) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 88
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory (AFAL/AA) Wright-Patterson AFB, Ohio 45433		12. REPORT DATE January 1977 13. NUMBER OF PAGES 82
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

High-Order Language
JOVIAL
Avionics Software
Avionics Standardization

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report documents the results of a high-order programming language (HOL) standardization study for avionics systems. Existing HOL usage was surveyed as well as current status and trends of Air Force and DOD HOL standardization attempts. An in-depth comparison was made of the status of the two most prominately used HOL in avionics; JOVIAL-73 and JOVIAL-3B. Trends in usage of HOL's and assembly language for avionics were discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

011 670

FOREWORD

In July 1976, an in-house effort was undertaken by the Air Force Avionics Laboratory to develop a laboratory-wide position for the use of higher-order languages (HOLs). The System Avionics Division (AFAL/AA) was tasked with developing this position and also tasked with developing a strategy whereby a "standard HOL" could be used if possible. In order to accomplish these overall tasks, an ad hoc team was formed and several separate and specific work-tasks were identified. The output of this effort is a series of three reports, of which this is the second. The other two reports are:

"Interim Report on the Status of the JOVIAL-73 and JOVIAL-3B Languages," August 1976.

"Report on Recommended HOL Policy for AFAL," September 1976.

For completeness, the information contained in the first report has been included in this report as Appendix B.

The author is grateful to the various government project personnel and contractor personnel who contributed their time and efforts to collecting much of the data on which these reports are based.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This report documents the results of a language standardization study performed by the System Avionics Division, Air Force Avionics Laboratory. The purpose of the effort was to survey the existing usage of HOL programming for avionics, assess the status of ongoing Air Force HOL standardization efforts, and, in turn, to recommend a policy for AFAL to cover future usage of HOL's.

The report is organized in five major sections, and the four remaining are summarized below:

a. Section II - Discusses the ongoing DOD, Air Force, and AFSC efforts to select and promote "standard" HOL's. The JOVIAL-73 language appears to be the direction that the Air Force will follow for such a standard.

b. Section III - The use of assembly language is discussed, and the trend is established for the use of HOL and assembly language in future avionics systems. HOL appears to play an increasingly important role.

c. Section IV - The use of HOL programming for avionics is discussed, and candidate languages are reviewed. The two most prominent candidates, JOVIAL-73 and JOVIAL-3B, are discussed and compared in detail. The JOVIAL-73 language appears to be the best choice of the two.

d. Section V - The summary remarks point to JOVIAL-73 as the most widely supported and used HOL "standard," and indicates that it is the best choice for AFAL selection as an internal standard.

During the course of this study, considerable confusion arose over what was meant by the term "avionic software." Many people interpret it in a very narrow scope to mean only the "flight software" used to perform such functions as guidance, fire control, navigation, etc. Still others interpret it more broadly to also include "support software" required to aid in the production and maintenance of the "flight software." In order to obviate this semantics problem, the definitions of Appendix A were devised. These definitions represent a general concensus, are fairly self-explanatory, and provide a clear partitioning of the major functions. In the context of this terminology, the focus of this report is on "Avionic Mission Software" and the use of HOL's to support this type of programming. The particular definition used is:

> Avionic Mission Software - Software which executes in the flight computer and performs direct mission related functions. An analogous and frequently used term is "embedded" software.

SECTION II

ONGOING HOL STANDARDIZATION EFFORTS

OVERVIEW:

This section presents the results of a review of the current DOD and Air Force HOL standardization efforts. For this review two primary inputs were considered:

a. Currently applicable documents reflecting HOL guidance for the Air Force and DOD, in general. The three primary documents considered were DOD Directive 5000.29, AFR 800-14, and AFR 300-10.

b. Conversations with key Air Force and DOD personnel that are actively involved in establishing these HOL policies.

DOD DIRECTIVE 5000.29:

The DOD Directive 5000.29, "Management of Computer Resources in Major Defense Systems," 26 April 1976, establishes the policy for management and control of computer resources in major DOD projects. In this sense the directive is very broad and addresses many aspects of both computer hardware and software; eg, planning, configuration management, requirements specification, programming languages, etc. On HOL issues the essence of DOD 5000.29 is:

a. An HOL will be used unless it can be demonstrated that an HOL will not be cost-effective over the life-cycle of the system.

b. An "approved list" of interim standard HOL's will be developed and published as DOD Instruction 5000.XX. Only HOL's on that list can be used unless it can be demonstrated that none are cost-effective over the life-cycle of the system.

c. Each DOD approved HOL will be assigned to a designated control agent who will be responsible for such activities as validating compliance of a compiler implementation with the standard language specifications, configuration control of the language, disseminating language information, compilers, etc.

As can be easily seen, the "approved list" is a key aspect of the above requirements. However, to date this approved list has not been finalized, but efforts are proceeding to do so. Early in 1976, the DOD requested each of the services (Air Force, Army, and Navy) to submit their recommendations of HOL's to be included in the DOD 5000.29 interim standards list. The table below contains the essence of these recommendations, by service. Currently the DOD committee is considering these recommendations and a final published "approved" list should be available before the end of 1976.

TABLE II-1
HOL RECOMMENDATIONS BY SERVICE

All Services	Air Force	Navy	Army
FORTRAN COBOL	JOVIAL-3 JOVIAL-73/I JOVIAL-3B2 JOVIAL-4	CMS-2	TACPOL

As seen from the above table the Air Force recommended four versions of JOVIAL to be used on future Air Force projects. The only restriction cited was regarding JOVIAL-4, which was specifically designated to be

used at the Satellite Control Facility, Sunnyvale, California. SAMSO has implemented a large amount of JOVIAL-4 software at Sunnyvale, and has thus justified further promoting the JOVIAL-4 language at that installation only. To the person experienced in the remaining three JOVIAL languages (J-3, J-3B/2, and J-73/I), it is clear that these three heavily overlap in capability with JOVIAL-73/I being somewhat more comprehensive.

AFR 800-14:

The AFR 800-14, "Management of Computer Resources in Systems," 12 September 1975, is a very broad document in addressing the acquisition and support of both hardware and software "embedded" in Air Force weapon systems. The major emphasis in the programming language area currently dictates that, "higher order languages are to be used to the maximum extent practical in the system under development." Although no specific HOL requirement is dictated beyond that statement (eg, no approved HOL list, etc), this document is to shortly be revised on this specific topic. Revision will be made to bring AFR 800-14 in-line with the DOD 5000.29 requirements, eg, requiring non-HOL programming to be specifically justified, requiring an HOL to be used from an "approved" list, etc. This guidance should be available by early 1977.

AFR 300-10:

The third important document concerning HOL usage is AFR 300-10, "Computer Programming Languages," 20 October 1971. This regulation applies to large data processing applications and dictates "standard" HOLs for use in these systems. Specifically the current version (20 Oct 1971) specifies the following:

a. JOVIAL-3 - Specified to be used for command and control

applications.

b. FORTRAN - Specified to be used for scientific applications.

c. COBOL - Specified to be used for other (business) applications.

This regulation is currently undergoing a revision process, and the draft update contains a list of five approved HOLs; none of which is to be limited to a particular application as in the old version of AFR 300-10. The five languages are JOVIAL-3, JOVIAL-73/I, FORTRAN, COBOL, and PL/1.

DISCUSSION WITH KEY PERSONNEL:

During the process of researching the current status of HOL standards, discussions were held with most DOD and Air Force personnel who are directly involved in the current wave of standardization attempts. The general consensus on the trends in this area can be summarized in the following statements:

a. Regarding JOVIAL-73/I and JOVIAL-3B, most considered JOVIAL-73/I as a superior language to other JOVIAL languages, and viewed it as "the" standard that the Air Force will tend toward.

b. JOVIAL-3B is viewed as being applicable only to the avionics arena.

c. FORTRAN and COBOL are viewed as having indelible positions in the general data processing arena, and thus viewed as complementary to JOVIAL-73.

d. The Common Language-76 (CL-76), or DOD-1 as it is alternatively known, is generally viewed as a long-term replacement for all interim standards except FORTRAN AND COBOL. Thus CL-76 would replace the JOVIALs, TACPOL, and CMS-2. However, CL-76 is viewed currently as an R&D effort that will not be truly available until the early 1980's.

Figure II-1 pictorially depicts the general direction that these standardization attempts are taking.

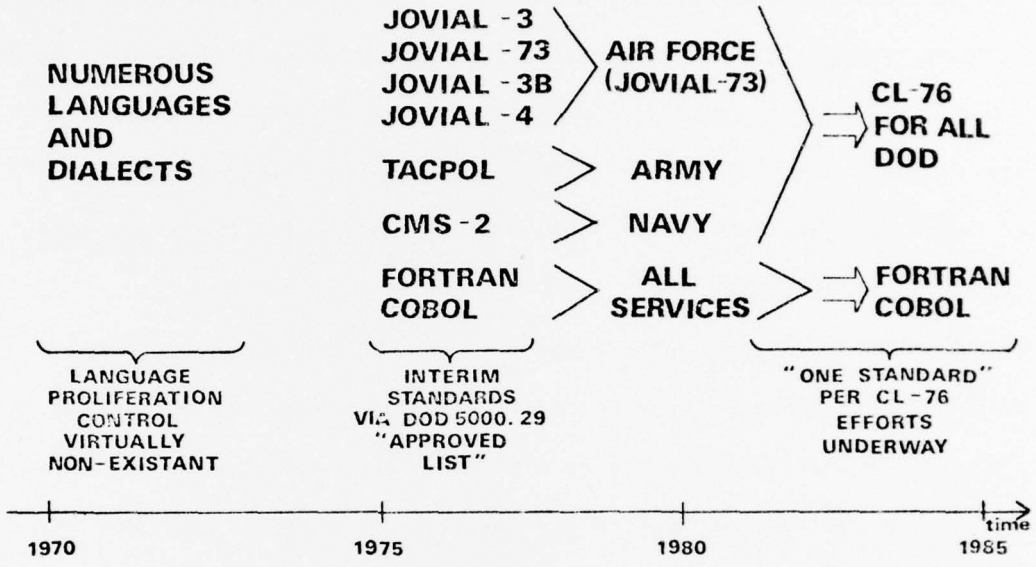


Figure II-1. Standardization Attempts

COMMON LANGUAGE-76:

The Common Language-76 (CL-76) effort has also been known as the DOD-1 language effort. It is being developed by DARPA as a long-term solution to the language proliferation problem and is to be applicable DOD-wide. The objective is to develop a single HOL that can replace the various procedure oriented HOL's (JOVIAL's, TACPOL, CMS-2, etc) that are currently being used and promoted by the three services. However, the current CL-76 effort is not being advertised as a replacement for FORTRAN and COBOL. Although no firm schedule exists for the CL-76 development, the general process is to be as follows:

a. A "TINMAN" requirements document now exists, and the various services have contracts with industry to compare these TINMAN requirements with 13 existing languages.

b. The outcome of these contracts (approximately November 1976) will be analyzed, and the future course of CL-76 will then be determined. Options seem to be: (1) modify existing languages or language; or (2) develop a new one from scratch. This plan should be available in April 1977.

c. Approximately two to three years from now a solid CL-76 language specification should exist.

d. Approximately five years from now a stable CL-76 compiler should exist, and the language be in a condition to be promoted throughout DOD.

Thus, CL-76 is viewed by most as a very long-term solution, and each service will be developing interim standards to be used for the next five years.

SUMMARY

From both the regulations reviewed and the discussion with the "key" personnel, the JOVIAL-73 language appears to be the direction the Air Force will take as a standard HOL. This summary is further substantiated by later sections of this report.

SECTION III

ASSEMBLY LANGUAGE ISSUES

GENERAL:

Assembly Language has been the traditional medium used for the development of embedded software for avionics systems. Even today, the majority of the avionic mission software being developed on current projects is being coded in assembly language. However, the trend is toward more and more use of HOL programming (particularly the JOVIALs) for a number of reasons. These are mainly:

a. The technology needed to produce highly optimized compilers has been steadily progressing, and state-of-the-art compilers are capable of producing machine code nearly (<10% expansion, average) as efficient as traditional assembly language programming.

b. With the introduction of "structured programming" and "software engineering" as a set of rigorous design and coding disciplines, HOL's can better compete, efficiency wise, with assembly language programming. Under these new "quality software" disciplines the assembly language programmer can no longer use the traditional "tricky" techniques which have always given him the edge over the HOL arena.

c. The use of digital processing in avionics systems has been increasing drastically over the recent past. This has resulted in large quantities of mission software being required, and in turn, requiring some methods or techniques to aid in the production and maintenance of this software. One method, the use of HOL, has been viewed as a key

ingredient in making the production of these large quantities of mission software a "manageable" process.

d. The trend in digital hardware has been toward drastically decreasing costs, and this has made the costs of both airborne and ground support computers much less than in the past. To this end, the inefficiencies in compiler-produced code are less of an impact on the total airplane system cost. Also, the costs of ground facilities needed for HOL compilers is likewise much reduced.

e. HOL's of a modern vintage, eg, JOVIAL-73, are much superior in capability to those of the previous era. For example, JOVIAL-73 can be used (and efficiently so!) for applications far beyond the traditional mathematical computations. Applications such as real-time executives, I/O interface programming, data base management, etc, are now possible in these modern HOL's.

ASSEMBLY LANGUAGE VERSUS HOL:

At the outset it should be made clear that, practically speaking, HOL's will probably never completely supplant assembly language programming for avionics. This is mainly because of management constraints and not because of technical considerations (such as language adequacy). For small avionic computers (eg, less than 4K memory), there will be instances where it will not be cost-effective to use an HOL. The general trend for avionics language usage is pictorially shown in figure III-1. What is indicated generally is that HOL's will indeed play an ever increasing role in avionics mission software.

The pressing question which needs addressing is to develop the

current trade-off analyses that govern whether or not an HOL can be cost-effective on a particular project. These analyses are very complex and involve many objective and subjective factors. For example, factors such as the costs of memory and CPU, cost of the compiler, schedule constraints, reliability/maintainability requirements for mission software, support computer availability, programmer training/abilities, etc, all have significant bearing on the choice of an HOL versus assembly language. It is generally agreed that for a large (more than 32K words of memory) avionics application, the use of an HOL is warranted as cost-effective. However at the other end

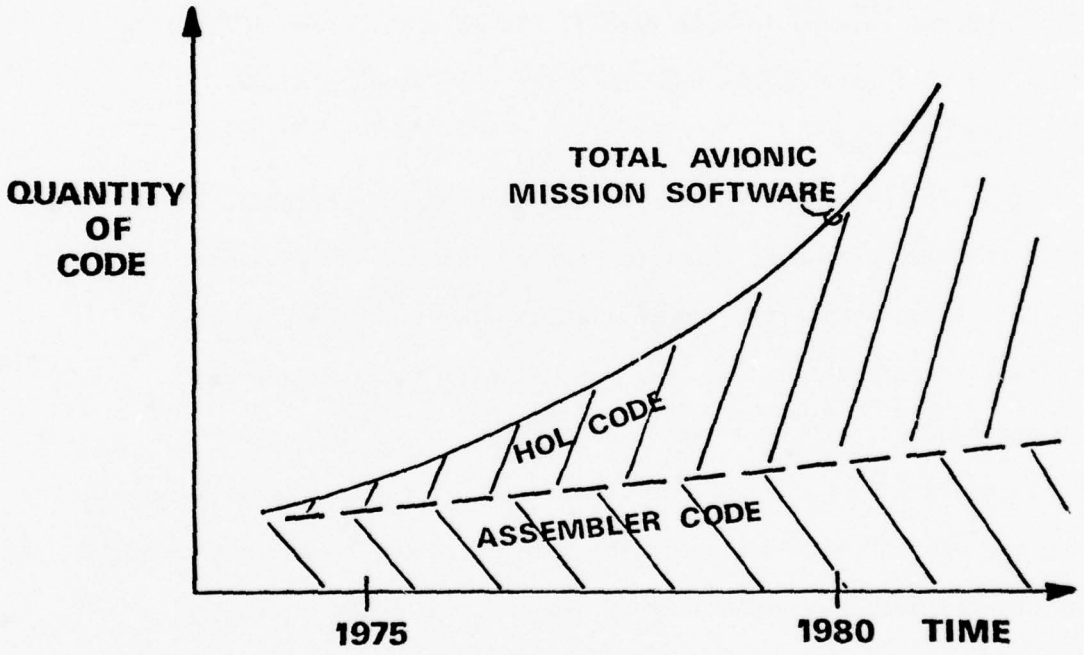


Figure III-1. Language Usage Trends for Avionics Mission Software

of the processing spectrum, the micro-processor HOL's are generally not considered to be cost-effective for many avionics applications. In between these two extremes of the avionics processing spectrum, there is an urgent need to establish the trade criteria mentioned above.

SECTION IV

HOL PROGRAMMING ISSUES

LANGUAGE CHOICES:

For application to avionics mission software programming, there are several viable candidate languages. These languages vary widely in capabilities, and also vary just as widely with respect to compiler efficiencies for the various implementations. A few of the more prominent candidates are discussed below, and in particular JOVIAL-73 and JOVIAL-3B are covered in some detail. To date, these two JOVIAL's have been the most extensively used HOLs for the avionics arena.

1. JOVIAL-73 LANGUAGE

History: The prime motivation for the development of JOVIAL-73 was the desire to have a common, powerful, easily understandable, and mechanically translatable programming language, suitable for a wide-range of Air Force applications. It was designed to be relatively machine independent, with a power of expression in logical operations and symbol manipulation, as well as numerical computation. Also, since the language was intended for application to large-scale data processing systems, it includes the capability of designating and manipulating system data as contained in a communication pool (COMPOOL).

JOVIAL-73 was designed by a combined committee of both industry and Air Force representatives, and was primarily targeted as a replacement for the outdated JOVIAL-3 Language Standard (AFM 100-24). The particular design goals of JOVIAL-73 were to satisfy the requirements of the following application areas:

-Avionics System
-Executive Writing
-Scientific Programming
-Tactical Systems
-Compiler Writing
-Data Management Systems
-Command and Control
-Real-Time Control

Of particular interest to this report is that of all the above applications, the first, avionics, received the most attention. This came about as the result of a survey of the avionics industry performed by the B-1 SPO in the Spring of 1972, in which both desirable and absolutely necessary language features and structure (for avionics) were asked for. Virtually all of the suggestions which fell into the realm of the programming language were incorporated into JOVIAL-73. Unfortunately, the suggestions had to be coordinated with the rest of the language forms and proposed structures which were desirable for the other applications, and due to time limitations the final specification could not be completed for use by B-1. Since that time the JOVIAL-73 specification (Ref #5) was completed and approved by the committee in November 1972, and published in January 1973.

In early 1973, a hierarchical system of subsets for JOVIAL-73 was developed (JOVIAL-73 Subsetting Report, dated 31 July 1973) to enable computer systems which cannot efficiently support the full JOVIAL-73 language to be able to implement a standard subset language and compiler. The most restrictive subset is supportive of the production of avionics systems, executives and operating systems, communication systems and other real-time control systems; compilation on computers having small to medium

scale main memory capacity; and, realization of programs on computer systems whose power and capacity is limited for their mission. An investigation to develop the hierarchical system of subsets was conducted as a benefit/cost trade-off analysis. To establish the benefit value of JOVIAL-73 features (i.e., language structures, computational structures, and the ways of realizing these), a relative utility value was identified for several hundred features within more than fifteen areas of application for the language. Additional relative benefit and cost factors were judged for many features. These included compiler development complexity, computer storage requirement for compilation, processing time requirement for compilation, difficulty of learning and successful use by programmers as costs; and, object program time and memory space efficiently utilized and programmer productivity as benefits. The analysis considered direct trade-offs between features and cost/benefit influence of the features. From this analysis there evolved three subsets forming a hierarchical system in which each subset was judged to be suitable for certain application areas; each subset took on a dominant characteristic. The base subset, JOVIAL-73/I was most suitable for programming in a very explicit manner, thereby providing greater control over the efficiency of the resultant object program. This subset was selected as the minimal desirable subset for a sufficiently wide range of applications that version proliferation would be minimized. In the final analysis, the three subsets were adjusted for consistency within each subset and to provide (still considering overall cost/benefit) that whatever computational capability could be realized in the full language could also be realized in every subset even though requiring a more arduous effort on the part of the programmer.

CURRENT STATUS:

In order to assess the current status of JOVIAL-73 usage, a survey was undertaken and the results are documented in Appendix B. Contained in this appendix is a description of each project using JOVIAL-73, how extensively the language has been used in each of these projects, and some disucssions on the status of compilers for JOVIAL-73. The highlights of these survey results are:

1. Approximately 569K words of JOVIAL-73 object code exists today, and an estimated 399K words more are either currently under development or shortly will be.

2. Seven companies presently have experience in the development of JOVIAL-73 code, and at least two others will shortly be involved in large JOVIAL-73 developments.

3. Eight major government organizations are involved in projects that currently are, or shortly will be, using JOVIAL-73.

4. Eight separate projects have used JOVIAL-73 to date, and another six projects are scheduled to shortly be users of JOVIAL-73.

5. The uses of JOVIAL-73 to date have been varied (avionics, real-time executives, compiler writing, and large scale simulations), and soon it will be used for multiple command and control applications in conjunction with the RPV project.

6. To date, two companies have produced JOVIAL-73 compilers, and one other is currently involved in the process of developing a compiler. Additionally, two other companies have had experience with maintenance of JOVIAL-73 compilers.

2. THE JOVIAL-3B LANGUAGE

History: When the RFP for the B-1 bomber system was prepared at Aeronautical Systems Division (ASD), it was decided to require the programming of the mission software to be done in an HOL. As ASD was an active member of the committee which designed JOVIAL-73 and was thus aware of its potential merit, in October 1971, JOVIAL-73 was selected as the language for B-1. However, as mentioned above, the JOVIAL-73 language was not ready in time, so an effort was made in June 1972 to devise a "bare-bones" HOL of low implementation risks which would satisfy the minimum needs of the B-1 until JOVIAL-73 became available. To achieve the low risk aspect, this language was designed as close as possible to JOVIAL-5, which is a subset of JOVIAL-3 and for which a Government-owned compiler already existed. The language was designed by Boeing and three members of the JOVIAL-73 Committee, namely RADC, ASD (The B-1 SPO), and ABACUS. So wherever possible, the language was made to look like what was known of JOVIAL-73 at the time.

The contract for the JOVIAL-3B compiler was finally awarded to a contractor (Softech) who did not propose the use of the JOVIAL-5 compiler as a baseline. Thus the need for staying close to JOVIAL-5 was eliminated. Since much more was known about the direction of JOVIAL-73 by the time the JOVIAL-3B specification was completed in October 1972 (by Boeing and Softech), it looked even closer to JOVIAL-73. Two compilers, one for the IBM-360 and one for the SKC-2000, both written in AED and residing on the IBM 360, were delivered to Boeing in September 1973.

This original "bare-bones" version of JOVIAL-3B has since been termed JOVIAL-3B/0, or version zero. The JOVIAL-3B/0 language and compiler for the SKC-2000 series computer were used for the development of the B-1 Offensive Flight Software. Later in the B-1 Project, a Defensive Subsystem was specified for use on the B-1 aircraft, and the decision was made to similarly code that software in JOVIAL-3B. However, the JOVIAL-3B/0 language was not sufficient for these new tasks, and was thus "upgraded" and expanded to the JOVIAL-3B/1 dialect (an upward compatible superset). Thus, the B-1 Defensive Flight Software (primarily on the LC-4516D computer) was implemented in JOVIAL-3B/1.

Later in the 1975 time frame, General Dynamics chose the JOVIAL-3B language for implementation of the fire control subsystem on the F-16 fighter aircraft. However, the JOVIAL-3B/1 language was again deemed insufficient and was "upgraded" and expanded to the JOVIAL-3B/2 dialect. This JOVIAL-3B/2 dialect is, in turn, a superset of JOVIAL-3B/1.

CURRENT STATUS:

In a manner similar to the survey conducted for JOVIAL-73, the current status of JOVIAL-3B was assessed, and the results are also contained in Appendix B. Each project using JOVIAL-3B is shortly discussed, and metrics are included to show how extensively the language has been used to date. Also included are some discussions of the status of compilers for JOVIAL-3B. The highlights of these survey results are:

1. Approximately 143K words of JOVIAL-3B object code exists today, and an estimated 50K words more are currently under development.

2. Four companies (Boeing, Softech, General Dynamics, and AIL)

have experience in development of JOVIAL-3B code.

3. Three major government organizations are involved in projects presently using JOVIAL-3B.

4. Four separate projects have, or are, using the JOVIAL-3B language.

5. The uses of JOVIAL-3B have been restricted to two applications areas; avionics and real-time executives. However, virtually all of this usage has been in the avionics area.

6. Only one company, Softech, has had any appreciable experience with building JOVIAL-3B compilers, however Boeing has had some experience with maintenance of the B-1 project compilers.

7. All of the current JOVIAL-3B compilers are implemented in the AED language, and this AED language is only supported by Softech. AED compilers and maintenance can be purchased from Softech, but Softech is the only vendor available.

3. OTHER HOL ALTERNATIVES

There are numerous other HOL's in existence which are possible alternatives for use in avionics. However, virtually all these have severe technical limitations that make them poor choices. For example, FORTRAN is many times discussed as a possible alternative. Indeed, FORTRAN is well suited to many of the mathematical computations required, but has few features that would permit efficient programming of input/output, bit and partial word data manipulation, intricate control logic, data packing and allocation control, address manipulation, etc. It is true that some of the above types of problems can be implemented in some FORTRAN's, but not in an efficient way. Since the FORTRAN language

constructs and data types are not intended for these operations, "tricky" programming techniques must be used to accomplish most of these. Such tricky techniques not only circumvent the language, but also "trick" the compiler to the extent that the optimizer cannot function well in producing efficient code. So, even if such problems could somehow be coded in FORTRAN, they would still be grossly inefficient.

Another language which has been used for some avionics applications is HAL/S. The language was developed by NASA for use on the Space Shuttle project, and has many features of "modern" programming languages. Also, the Air Force Avionics Laboratory, Global Positioning Satellite/Generalized Development Model (GDM) Project is using HAL/S for its prototype flight software development. However, HAL/S suffers from the lack of any large user base or the support for standardization that would make it a cost-effective choice for most projects.

There have been many other languages discussed, and sometimes used, for avionics (eg, CMS-2, AED, PL/1, JOVIAL-3, SPL, etc), but all of these suffer from either severe technical or management problems. Thus to date the only serious candidates for use in the avionics area have been JOVIAL-73 and JOVIAL-3B. References 1 through 4 are a few of the major documents addressing HOL comparisons for avionics.

HOL VERSUS HOL

As alluded to in earlier sections, there are many factors which affect the HOL selection decision. Some of these are "technical" factors, such as those which contribute to making the language "adequate" for the job at hand. However, there are also numerous "management" oriented factors

which must be equally considered; for example, the maturity of the language, the size of the user base, availability of compilers, tools and documentation, etc. Each of these factors will be discussed below, and an assessment will be made of the status of both JOVIAL-73 and JOVIAL-3B relative to each factor. Due to the severe limitation of HOL's other than JOVIAL-73 and JOVIAL-3B, only these two will be further analyzed in this report.

1. TECHNICAL FACTORS/LANGUAGE ADEQUACY: The first consideration in any HOL selection process is the adequacy of the language to do the particular job. Over the past few years there have been numerous studies conducted which have compared HOL's for use in avionics, and some of these are identified in the reference section. Within the scope of this report the detailed comparison of language capabilities will not be attempted; instead, it will suffice to say that a language with capabilities similar to JOVIAL-73/I is adequate and well suited to avionics. Indeed, this is the general synoposis of most studies in this area.

In comparing JOVIAL-73 and JOVIAL-3B from the "adequacy" viewpoint, the following general statements can be made:

a. JOVIAL-73/I is a superior language to both JOVIAL-3B/0 and JOVIAL-3B/1.

b. The latest dialect of JOVIAL-3B, that is JOVIAL-3B/2 as used on the F-16, is comparable in capability to that of JOVIAL-73/I.

The essence of these statements is that through the JOVIAL-3B evolution process (i.e., JOVIAL-3B to JOVIAL-3B/1 to JOVIAL-3B/2) a language of the level of JOVIAL-3B/2 or JOVIAL-73/I has been shown to be proper for avionics applications. Also, since JOVIAL-3B/2 and JOVIAL-73/I are comparable in capabilities, the "adequacy" factor can be normalized out in trade studies between these two languages.

2. MANAGEMENT FACTORS: Since JOVIAL-73/I and JOVIAL-3B/2 are so similar in capabilities, the choice between these two HOL's is centered on the more management oriented considerations. These are discussed and compared below.

a. Higher Headquarters Directions: In order to remain in compliance with guidance and regulations documents, the current and projected directions from higher headquarters organizations must be understood and considered. As discussed in detail in Section II, there appears to be strong support for the promotion and standardization of JOVIAL-73 over JOVIAL-3B. In particular, the current draft version of DODI-5000.XX contains JOVIAL-73 as a standard, not JOVIAL-3B.

b. Size of User Community: It is advantageous for a project to select a language that has a large base of users. This allows the project to capitalize on the past, and future, investments which others have made in the HOL; for example, compilers, documentation, training, test tools, etc. To summarize some of the data presented in previous sections, Table IV-1 compares JOVIAL-73 and JOVIAL-3B on several major factors. From such data it is easily seen that JOVIAL-73 is used more extensively today than JOVIAL-3B and is projected for an even larger portion of the future Air Force business.

c. Availability of Compilers: Since most projects do not plan for HOL's far enough in advance to afford the time needed for a compiler development, the availability of existing compilers becomes a very important factor. Again the previous sections presented data on both JOVIAL-73 and JOVIAL-3B compilers for various machines. The key items are also summarized in Table IV-1. Although the numbers of compilers are comparable between the two HOL's, an important difference must be noted;

TABLE IV-1

SUMMARY COMPARISON DATA

PARAMETER		METRIC	LANGUAGE			
			JOVIAL-73		JOVIAL-3B	
NUMBER	Description		To Date	Additional Near Term	To Date	Additional Near Term
1	Language usage extent	Words of object code produced from JOVIAL Source	569K *	399K	143K	50K
2	Contractors involved in language usage	No. of companies	7	2	4	0
3	Major government agencies involved in language usage	No. of agencies	5	3	3	0
4	Projects using language	No. of projects	8	6	4	0
5	Various applications for which language used	No. of different applica. areas	4	1	2	0
6	Experienced compiler vendors for language	No. of companies	3	0	1	0
7	Experienced compiler "Maintainers" for language	No. of companies	5	0	2	0
8	"Production" class compilers	No. of compilers	2	2	5	0
9	Total compilers	No. of compilers	5	2	5	0
10	Different Hosts for "Production" compilers	No. of Host computers	1	1	1	0
11	Different Hosts for all compilers	No. of Host computers	3	1	1	0

*Largely Avionic Support Software

presently there is not a production level JOVIAL-73 compiler available on the IBM-360 computer. Due to the wide usage of IBM-360's, this can indeed be a limiting factor. However, plans are made for hosting the JOVIAL-73/I DAIS compiler on an IBM-360.

d. Level of R&D Support: A key element to ensuring that a language is accepted by a user community and fulfills their needs is to develop not only the language itself, but to also develop a comprehensive set of tools and other aids to support the language. This includes such tools as validators to ensure that various compilers properly implement the language, compiler building tools to allow for the fast production of compilers, debugging aids to assist programmers in the testing of code, etc. Since most of these tools fall within the realm of R&D, it is thus important for a potential language user to assess at the level of this R&D support.

In reviewing JOVIAL-73 and JOVIAL-3B relative to this R&D support the following results were found:

a. There are no active efforts underway for R&D support to the JOVIAL-3B language. In addition, there is no organization with plans to sponsor R&D for this language.

b. For the JOVIAL-73 language, RADC has planned and is currently sponsoring several R&D activities for this language. In addition, RADC has planned increasing investments in JOVIAL-73 tools for the near future. Appendix C summarizes these major efforts.

SUMMARY:

Table IV-2 summarizes the status of JOVIAL-73/I relative to JOVIAL-3B/2. As is seen from this data the JOVIAL-73/I language appears to be more solidly supported by users, the policy makers, and the R&D community. The one weak point of JOVIAL-73/I is the lack of a compiler on an IBM-360 series computer. However as mentioned in earlier sections, plans are being finalized to shortly put the DAIS Project JOVIAL-73/I compiler on the IBM 360/370 computers.

TABLE IV-2

JOVIAL-73/I AND JOVIAL-3B/2 STATUS SUMMARY

FACTOR	COMMENTS	
	JOVIAL-73/I	JOVIAL-3B/2
Language Adequacy	Comparable with JOVIAL-3B/2	Comparable with JOVIAL-73/I
Higher Hq Directions	Strongly favored over JOVIAL-3B/2	
Size of user community	Larger than for JOVIAL-3B/2	
Availability of Compilers	"Production" compilers on DEC-10 and U1110. IBM 360/370 compiler planned.	Advantage: "production" JOVIAL-3B compiler currently on IBM-360
Level of R&D Support	Moderate support	None

SECTION V

SUMMARY AND RECOMMENDATIONS

SUMMARY

1. The use of HOL programming for avionics is currently well established, and will be even more heavily used in future systems. This trend is due to two main factors:

a. The improved capabilities of today's modern HOL's and the increasing cost-effectiveness of these HOL's compared to assembly language.

b. The increased emphasis within DOD and the Air Force to utilize HOL's on all major weapon system projects.

2. There is a current emphasis within both DOD and the Air Force to reduce the proliferation of HOL's. To this end, "standard" HOL's will be specified for use in future systems.

3. Of the HOL's possible for use in avionics, JOVIAL-73 and JOVIAL-3B are by far the most prominent. Of these two, JOVIAL-73 is more strongly supported as a standard within the Air Force, and also more extensively used for total avionics applications (support software plus mission software). However, most of the use of JOVIAL-73 to date has been for avionic support software.

RECOMMENDATIONS:

In light of the results compiled in the previous sections, the following recommendations are made.

1. Since both HOL and assembly language programming will continue to share the workload in avionics programming, a set of "guidelines"

should be developed to aid in the choice between the two. These guidelines should identify and quantify the appropriate trade-off factors needed in these decisions.

2. AFAL should develop a set of guidelines to govern the selection and use of HOL's on future projects. The basis of these guidelines should be the use of the JOVIAL-73/I standard language. In essence, future projects should not be allowed to use an HOL other than JOVIAL-73/I.

SECTION VI

REFERENCES

1. Boeing, Higher Order Language Application Study, JOVIAL-73, July 1973.
2. General Dynamics, Definition Study of a Digital Avionics Information System (DAIS), October 1973.
3. Texas Instruments, Definition Study of a Digital Avionics Information System (DAIS), October 1973.
4. General Dynamics, Operational Software Concept (Phase II), AFAL-TR-75-230, January 1976.
5. Standard Computer Programming Language JOVIAL (J73), January 1973.
6. DAIS Specification SA204200, JOVIAL (J73/I) Language Specification, December 1974.
7. L. Shirley, JOVIAL-73 Subsetting, Air Force RADC Report, July 1973.
8. AFR-300-10, Computer Programming Languages, October 1971.
9. DOD Directive 5000.29, Management of Computer Resources in Major Defense Systems, April 1976.
10. AFR-800-14, Acquisition and Support Procedures for Computer Resources in Systems, September 1975.

APPENDIX A

AVIONICS SOFTWARE TERMINOLOGY

The term "avionic software" is frequently used in many different ways and to mean vastly different concepts. Thus in order to aviod confusion within this report, some basic definitions are first made to identify what is included in the term "avionic software." Figure A-1 expresses the relevant terminology in a "tree" format showing these relationships. The first distinction to be made is between; (a) actual flight code that performs the aircraft mission and test functions (eg, resides in the flight computer), and (b) the miriad of support software

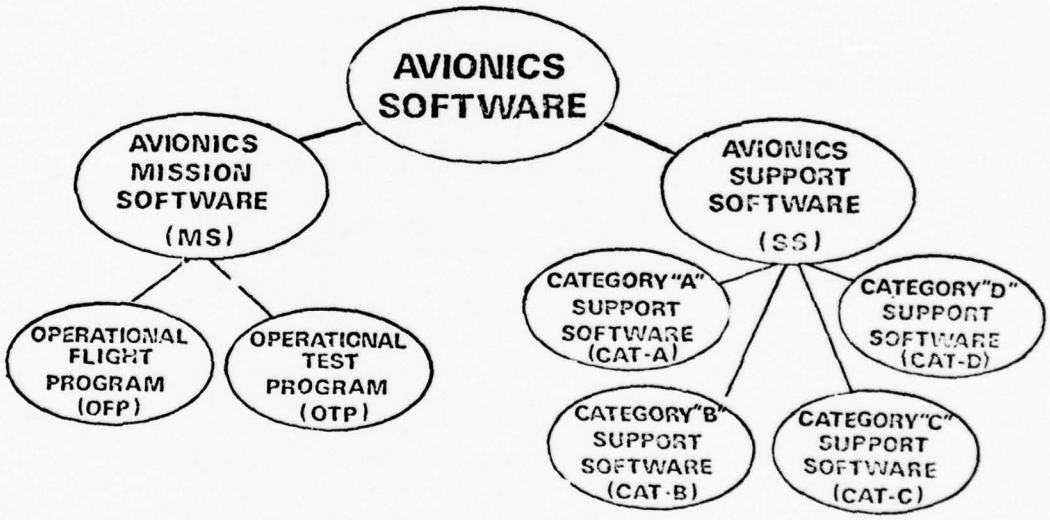


Figure A-1. Terminology Tree

that is needed to aid in the design and development of an avionic system. Some working level definitions are:

Avionic Mission Software (MS): Software which executes in the flight computer and performs direct mission related functions. An analogous and

frequently used term is "embedded" software.

Avionic Support Software (SS): Software which is required to aid in the design and production of avionic systems (both hardware and mission software).

However these terms are still very broad in scope, and a third level of categorization is needed. For Avionic Mission Software, a rather clear distinction can be made between real-time flight code (Operational Flight Program) and the flight-line loadable system test software (Operational Test Program). The appropriate definitions are:

> Operational Flight Program (OFP): The OFP contains those functions which execute in the flight computer in real-time (or near real-time) and perform the primary aircraft mission functions. Examples for a fighter type aircraft are; executive control, navigation, guidance, stores management, electronic warfare control, targeting/fixtaking, controls/displays processing, communications, in-flight systems test, etc.
>
> Operational Test Program (OTP): For most modern day aircraft systems, a comprehensive test program is needed for flight-line maintenance of the avionics suite. The OTP usually takes the form of a flight-line loadable software package that will execute in the actual flight computers while these computers are still resident in the aircraft. The results of executing the OTP will be the identification of avionic system problems, isolation of these problems to faulty units, and/or certification of the avionic system hardware to be free of failures and suitable for a mission. Example functions of the OTP include; computer health tests, I/O hardware tests, sensor operation tests, fault isolation and unit identification.

Also, the Avionic Support Software term involves many different "types" of software, and a further categorization is needed here. Not too creatively, names were coined such as "Category A," "Category B," etc. Each of these are discussed below.

> Category "A" Support Software (Cat-A): This category contains those major support software functions that operate in non-real-time (ie, off-line) and directly

support the development of the Mission Software. Examples of Cat-A functions are; high-order language compilers, assemblers, linkers/loaders, off-line simulations such as interperative computer simulators, file management aids for configuration management of mission software modules, debugging aids, etc. These programs generally operate on a support computer at a maintenance or production facility.

Category "B" Support Software (Cat-B): This category contains the major support software functions that operate in real-time and support the development and validation of the Mission Software OFP and OTP. Also, since the real-time support software needed for flight trainers is so similar, this software is also included. Cat-B software will generally operate on a support computer at a maintenance or production facility, or in a training facility. Example functions include; simulation models of avionic sensors, real-time executives, models of interface hardware, real-time debugging aids, real-time data collection, test control and test operator interface, mission senario generation, training aids, etc.

Category "C" Support Software (Cat-C): This category contains the major support software functions that operate in non-real-time and directly support the reduction, analysis and generation of mission flight data. An example would be the support software needed to generate a data base of electronic countermeasures tactics to be used on a particular mission of a fighter aircraft. Another example would be the support software needed to generate navigation data-tapes containing pre-programmed way-points, radio navigation station usage data, threat areas to aviod, etc. Again, this software is usually resident on a support computer at a development, maintance, or operational support facility.

Category "D" Support Software (Cat-D): Unlike categories A, B, and C which have some direct connection with the production, operation, or maintenance of the Mission Software, the Cat D Support Software is much further removed from the flight environment. This category contains those large support software "systems" that have use as off-line design aids for the early stages of avionics system developments. A typical example would be large avionic mission/system simulation such as the Avionic Evaluation Program (AEP).

Figure A-2 depicts these avionic software categories in another perspective - - that of the computer in which the software is resident.

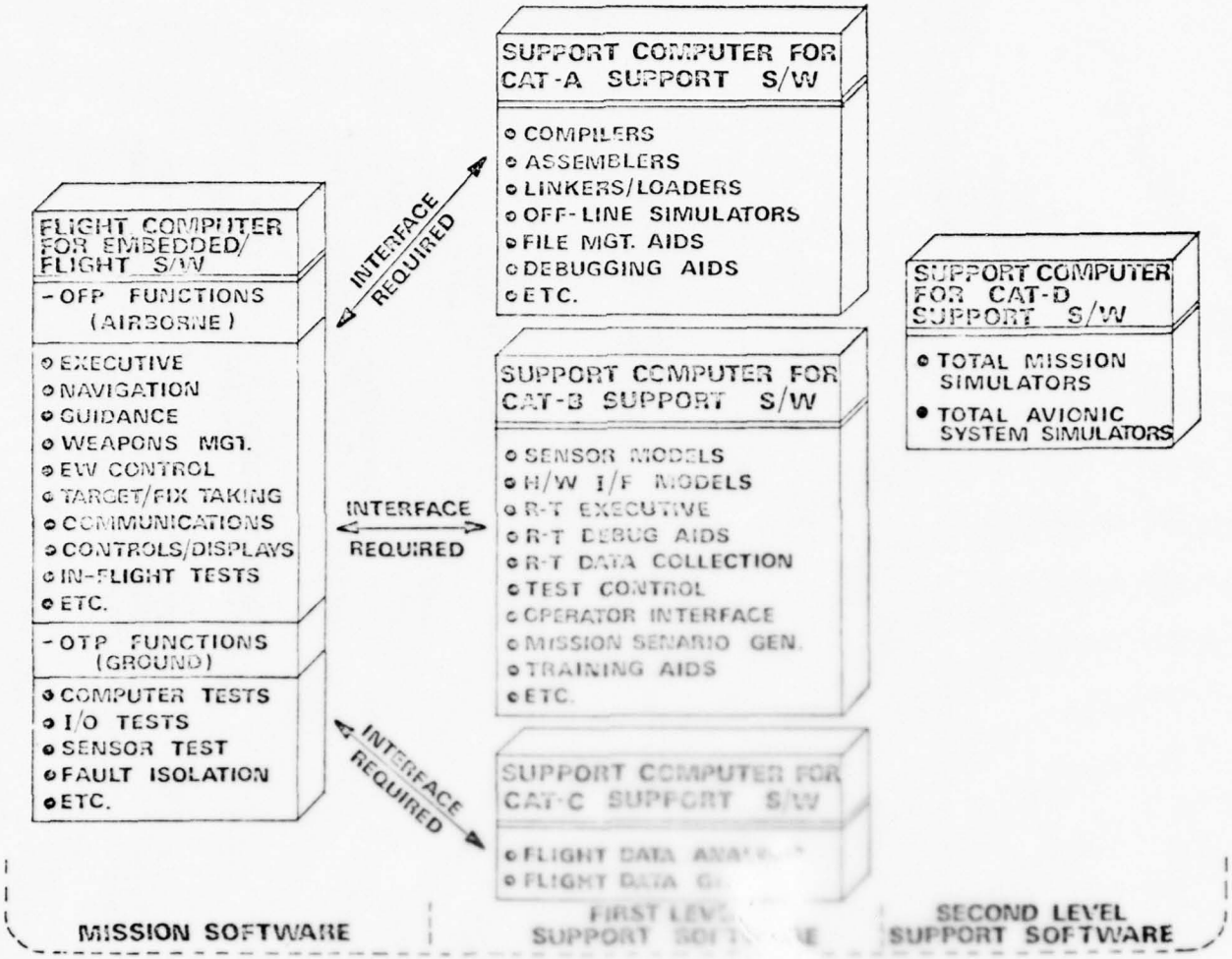


Figure A-2. Categories of Avionics Software

The Mission Software (OFP and OTP) are resident on the airborne computer, and is shown at the left. If one views the mission to be accomplished (eg, a tactical aircraft strike mission) to be at the left margin of this figure, the Mission Software is indeed the "closest" and contributes the most directly to the accomplishment of that mission. At the next level to the right is viewed the first level of avionic Support Software: Cat-A, B, and C. These three categories are all thus further removed from the mission and flight vehicle itself. Finally at the right of this figure is viewed the Cat D Support Software which contributes least directly to supporting the particular mission.

Figure A-3 is yet another way of pictorially presenting this concept of "layered" software showing the flight computer and mission at the center of the figure. Software further from the center is less directly involved in mission accomplishment.

Also, as drawn in figure A-2 the four categories of avionic Support Software are shown in four "conceptually" different computers. In actual practice one will find that at least four, and usually more, different support computers will be used for any given aricraft system. This is usually not dictated by technical requirements since there are projects that have used as few as two different support computers, but is usually more as a result of poor system engineering practices.

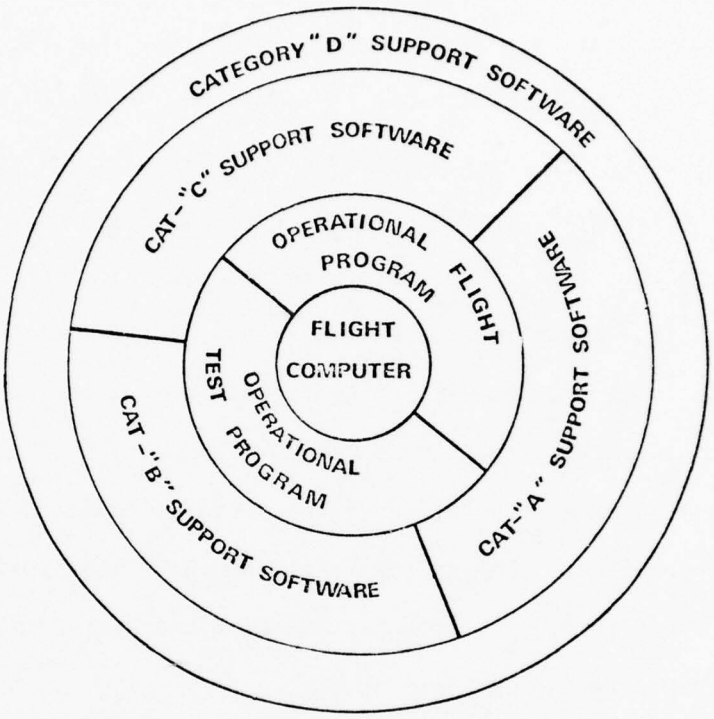


Figure A-3. Avionics Software Terminology

APPENDIX B

JOVIAL-73 AND JOVIAL-3B STATUS COMPARISON

TABLE OF CONTENTS

SECTION I

INTRODUCTION

This appendix was generated to document the results of an effort undertaken by the System Avionics Division, Air Force Avionics Laboratory, to compare the status of the JOVIAL-73 and JOVIAL-3B programming languages. Both languages are high-order computer programming languages which were developed for application on Air Force weapon systems, in fact, JOVIAL-3B is in reality a derivative of the original JOVIAL-73 specification process. The data presented in the subsequent sections represent the results of a survey which involved queries of many government organizations and contractors. Attempts were made to make the survey as complete and accurate as possible. In some instances gross estimates were the only data available - these instances are so noted in the charts and texts.

A special note is in order concerning the data relating to "language usage" (ref. Sections II and III). Efforts originally were directed at collecting the language usage data in terms of "lines of JOVIAL source code"; a somewhat better metric than "words of object code produced from JOVIAL source." However in many cases the former type of data was not readily available, and projects did not have the excess resources that would have been needed to collect such data. Therefore, in the interest of obtaining a more complete survey, all data were finally recorded as "words of object code produced from JOVIAL source."

The report is organized in six major sections. Sections II and III contain data on usage of JOVIAL-73 and JOVIAL-3B, respectively. Sections IV and V are complimentary to these and contain information on the compliers for JOVIAL-73 and JOVIAL-3B, respectively. Finally, Section VI is a short section summarizing the results. The final conclusion is that JOVIAL-73 is both a more "mature" language and possesses a larger base of users than does JOVIAL-3B.

SECTION II

JOVIAL-73 USAGE DATA

This section documents and summarizes the results of the survey of JOVIAL-73 language usage. TABLE (B)II-1 itemizes the results on the sixteen projects that JOVIAL-73 has either been, or shortly will be, used on. Each of these projects is covered in more detail in the subsequent pages. The salient points to note concerning these results are:

1. Approximately 569K words of JOVIAL-73 object code exists today, and an estimated 399K words more are either currently under development or shortly will be.

2. Seven companies presently have experience in the development of JOVIAL-73 code, and at least two others will shortly be involved in large JOVIAL-73 developments.

3. Eight major government organizations are involved in projects that currently are, or shortly will be, using JOVIAL-73.

4. Eight separate projects have used JOVIAL-73 to date, and another six projects are scheduled to shortly be users of JOVIAL-73.

5. The uses of JOVIAL-73 to date have been varied (avionics, real-time executives, compiler writing, and large scale simulations), and shortly it will be used for multiple command and control applications in conjunction with the RPV project.

6. To date two companies have produced JOVIAL-73 compilers, and one other is currently involved in the process of developing a compiler. Additionally, two other companies have had experience with maintenance of JOVIAL-73 compilers.

TABLE (B)III-1

CURRENT AND PLANNED J-73 USERS

GOVERNMENT AGENCY	PROJECT	COMPANY	PROGRAM SIZE (Words Memory)		STATUS
			Developed	Proposed	
AFAL/AA	DAIS Mission Software	Intermetrics	10K	70K	Under development
	DAIS Support (SDVS)	TRW	250K	0	Completed
	DAIS DEC-10 Compiler	CSC*	170K	0	Completed
	DAIS HBC Compiler	CSC*	35K**	0	Completed
	DAIS PALEFAC Software	CSDL	25K	17K	Under development
	Operational Software Concept	Softech	16K	Unknown	Under devleopment
AFAL/RW	Electronically Agile Radar (EAR)	Westinghouse	0	100K	Under development
AFATL/DL	DAIS Stores Management	In-House	0	10K	Under development
AFFDL/FG	DAIS Fly-By-Wire	In-House	0	16K	Under development
ASD/XR	Advanced Strategic Air Launch Missile	Martin-Marietta	1K	0	Completed***
ASD/YM	COMPASS COPE Flight Software	Unknown	0	6K	Source Selection
	COMPASS COPE GCCS	Unknown	0	65K	Source Selection
	RPV Mission Control System	Unknown	0	80K	RFP in writing
SAMSO/DY	Fault Tolerant Space Computer Compiler	CSC	0	35K	Under development
AFSC/CCPC	Language Evaluation	In-House	Unknown	Unknown	In process
RADC/IS	JOVIAL Compiler Validator System	Abacus	62K	0	Completed

NOTES: *This software developed by CSC but currently being maintained by TRW and Abacus.
**This is only memory requirements above basic compiler.
***This was a small subset of J-73/I.

PROJECT: DAIS Mission Software

GOVERNMENT AGENT: AF Avionics Laboratory/AA

DESCRIPTION: As a portion of the DAIS Project, the AF Avionics Lab, Wright-Patterson AFB, is developing the Mission Software (or "flight" software) for the DAIS Integrated Test Bed. This effort is being conducted under contract and is using AN-AYK-15 airborne computers. A MIL-STD-1553A data bus communication scheme is being used for sensor data communications, and also for inter-computer communications.

MAJOR SOFTWARE FUNCTIONS:

- Executive Control
- Bus control
- Equipment Interfaces
- Controls/Displays
- Inertial Navigation
- Doppler Navigation
- TACAN navigation
- Weapon delivery
- Instrument landing
- Target/fix taking
- Radio control
- Autopilot/steering
- Mode control
- Error control/reconfiguration

STATUS: This effort is presently underway. Portions of the local executive and navigation are coded and unit tested, and integration of these functions is presently ongoing.

SIZE OF EFFORT: The final software will be approximately 80K words. To date, approximately 10K words of code have been developed.

COMPANIES INVOLVED: Intermetrics, Inc., Cambridge MA

PROJECT: DAIS Software Design & Verification System

GOVERNMENT AGENT: AF Avionics Lab/AA

Description: As a portion of the DAIS Project, the AF Avionics Lab, Wright-Patterson AFB, has developed a comprehensive support software system to be used as an aid in developing the DAIS flight software. This Software Design & Verification System, SDVS, contains both simulation aids (simulators, test control languages, analysis/data reduction programs, etc.) and management aids (on-line data base, configuration control, etc.). This software is implemented on a DEC-10 computer system.

MAJOR SOFTWARE FUNCTIONS:

- Executive Control
- Statement Level Simulator
- Interpretive Computer Simulator
- Data bus simulator
- Simulation control
- Post-Run Analysis
- Linker/Loader
- Snapshot/Rollback
- Scenario generation
- File generation

STATUS: This effort is completed. A maintenance/enhancement effort is currently underway for continued support of this software.

SIZE OF EFFORT: The final software, delivered and operational, is approximately 250K words generated from JOVIAL-73 source code.

COMPANIES INVOLVED: TRW, Redondo Beach, CA

PROJECT: DAIS JOVIAL-73 Compilers

GOVERNMENT AGENT: AF Avionics Lab

DESCRIPTION: As a portion of the DAIS Project, the AF Avionics Lab, Wright-Patterson AFB, has developed two JOVIAL-73/I compilers; one for the DEC-10 computer and one for the AN-AYK-15 flight computer. These compilers were developed to support coding of both the DAIS flight software and support software. The compilers are also written in J-73. Both compilers were developed under contract.

MAJOR SOFTWARE FUNCTIONS:

- Compiler Executive
- Code Analyzer
- Allocator
- Code Generator
- Editor
- Optimizer

STATUS: This effort is completed. A maintenance/enhancement effort is currently underway for continued support of this software.

SIZE OF EFFORT: The final software, delivered and operational, is approximately 170K words for the DEC-10 compiler, and approximately 35K words additional for the AN-AYK-15 cross-compiler.

COMPANIES INVOLVED: Computer Sciences Corp., El Segundo, CA, was the original developer for both compilers. TRW, Redondo Beach, CA, with a subcontract to Abacus Corp., Santa Monica, CA, is currently performing the maintenance and enhancement of this software.

PROJECT: DAIS PALEFAC Software Development

GOVERNMENT AGENT: AF Avionics Lab/AA

DESCRIPTION: As a portion of the DAIS Project, the AF Avionics Lab, Wright-Patterson AFB, is currently developing a software program entitled PALEFAC. The PALEFAC function is a part of the overall DAIS support software system, and will automatically produce/validate executive and bus control tables needed for the DAIS flight software executive. This effort is being developed under contract and written in JOVIAL-73.

MAJOR SOFTWARE FUNCTIONS:

- Executive Table Generation
- Data Bus Table Generation

STATUS: Under development.

SIZE OF EFFORT: Approximately 25K words of code have been developed to date. Approximately 17K words remain to be developed.

COMPANIES INVOLVED: C. S. Draper Lab, Cambridge MA

PROJECT: Operational Software Concept

GOVERNMENT AGENT: AF Avionics Lab/AA

DESCRIPTION: The Operational Software Concept (OSC) is an advanced development effort in the AF Avionics Lab, Wright-Patterson AFB. This is a multi-phased effort to develop and demonstrate avionic flight software design and implementation methodologies which will help reduce the life-cycle-cost of flight software. This project consists of both contractual and in-house efforts, and to date has produced a prototype flight executive and limited applications software. Current OSC efforts are using JOVIAL-73.

MAJOR SOFTWARE FUNCTIONS:

- Executive Control
- Bus Control
- Air Data Computations

STATUS: The executive software has been developed in JOVIAL-73 on the DEC-10 computer system. An Air Data applications module has also been developed in JOVIAL-73.

SIZE OF EFFORT: The Executive consists of approximately 15K words of code. The Air Data computation modules comprise approximately 1K words of code. Additional executive and applications software are under development, but estimates of final size are not available.

COMPANIES INVOLVED: Softech, Waltham MA

PROJECT: Electronically Agile Radar

GOVERNMENT AGENT: AF Avionics Lab/RW

DESCRIPTION: The AF Avionics Lab, Wright-Patterson AFB, is currently under contract to develop the Electronically Agile Radar (EAR) system. This system utilizes a multi-mode, phased-array radar technology, and dually-redundant Westinghouse flight computers. The flight software is to be largely coded in JOVIAL-73.

MAJOR SOFTWARE FUNCTIONS:

- Navigation Update
- Terrain Following
- Antenna Beam Control
- Built-in-Test
- ECCM
- Mapping
- Fault-Isolation
- Controls/Displays

STATUS: This effort is currently underway and a B-5 software specification exists.

SIZE OF EFFORT: Approximately 100K words of code will be written in JOVIAL-73

COMPANIES INVOLVED: Westinghouse Corporation, Baltimore MD

PROJECT: DAIS Stores Management Mission Software

GOVERNMENT AGENT: AF Armament Test Lab/DL

DESCRIPTION: As a portion of the DAIS Project, the AF Armament Test Lab, Eglin AFB, is implementing the Stores Management functions to be integrated into the DAIS flight software. This effort is being conducted in-house and is using the AN-AYK-15 airborne computer. MIL-STD 1553A multiplex remote terminals for stores stations are also being developed at AFATL and will be integrated with the final DAIS Stores Management software.

MAJOR SOFTWARE FUNCTIONS:

- Weapon C/D Processing
- Weapon control
- Weapon status monitoring
- Weapon Inventory Control
- Post Flight Analysis

STATUS: This effort is currently under development. A draft B-5 specification exists at this time.

SIZE OF EFFORT: Final software size will be approximately 10K words.

COMPANIES INVOLVED: None, only in-house personnel.

PROJECT: DAIS Digital Flight Controls

GOVERNMENT AGENT: AF Flight Dynamics Lab/FG

DESCRIPTION: As a portion of the DAIS Project, the AF Flight Dynamics Lab, Wright-Patterson AFB, is implementing a quad-redundant digital fly-by-wire flight control system. This mechanization uses AN-AYK-15 airborne computers and a MIL-STD-1553A intercommunications scheme. This effort is being conducted in-house.

MAJOR SOFTWARE FUNCTIONS:

- Executive control
- Bus control
- Controls/displays management
- Mode control
- Inner-loop stability
- Outer-loop modes

STATUS: This software is presently under development, and prototype functions now exist on PDP-11 computers in assembly language. Final coding in J-73 for the AN-AYK-15 will begin next quarter.

SIZE OF EFFORT: The final software size will be approximately 16K words in each of four redundant computers.

COMPANIES INVOLVED: None, only in-house personnel.

PROJECT: Advanced Strategic Air Launch Missile

GOVERNMENT AGENT: Aeronautical Systems Division/XR

DESCRIPTION: As a portion of the ASD/XR project to design the Advanced Strategic Air Launch Missile (ASALM), the autopilot functions were prototyped in JOVIAL-73. This effort was conducted under contract and used the Air Research RF-20 computer.

MAJOR SOFTWARE FUNCTIONS:

- Bank-to-turn autopilot
- Attitude control
- Acceleration control

STATUS: Complete

SIZE OF EFFORT: The final software was approximately 1K words.

COMPANIES INVOLVED:

Martin-Marietta Aerospace, Orlando FL, was the ASALM contractor.

Proprietary Software Systems, Inc., Beverly Hills CA, developed the JOVIAL-73/M (a subset of JOVIAL-73/I) compiler for use by Martin-Marietta.

PROJECT: Compass Cope

GOVERNMENT AGENT: Aeronautical Systems Division/YM

DESCRIPTION: The Compass Cope project is a contractual effort managed by the Aeronautical Systems Division, Wright-Patterson AFB. This effort is for the design and development of a Compass Cope remotely piloted vehicle (RPV) and a ground control system. The Compass Cope RPV will have a flight computer and software that may be coded in JOVIAL-73, and will have a Ground Control Communications System (GCCS) that will be coded in JOVIAL-73

MAJOR SOFTWARE FUNCTIONS:

(1) Flight Software

- Navigation
- Steering
- Communications
- Fault Detection
- Reconfiguration
- Sensor control

(2) Ground Control

- Flight Planning
- Data Base Management
- Communications
- RPV control
- Navigation
- Controls/Displays
- Mission Analysis
- Executive Control

STATUS: Contract in source selection phase.

SIZE OF EFFORT:

Flight Software approximately 6K words

Ground Computer approximately 65K words

COMPANIES INVOLVED: Unknown

PROJECT: RPV Mission Control System

GOVERNMENT AGENT: Aeronautical Systems Division/YM

DESCRIPTION: The RPV Project Office of Aeronautical Systems Division, Wright-Patterson AFB, is developing a generalized ground command and control system for use with various types of remotely piloted vehicles (RPV's). This system, termed RPV Mission Control System (RMCS) will control many RPV's simultaneously and of different types. The RMCS software will be written in JOVIAL-73.

MAJOR SOFTWARE FUNCTIONS:

- Executive Control
- Mission Planning
- Navigation
- Data Base Management
- RPV Control
- Steering
- Communications
- Flight Planning
- Controls/Displays
- Mission Analysis

STATUS: RFP presently being written.

SIZE OF EFFORT: Approximately 80K words.

COMPANIES INVOLVED: Unknown

PROJECT: Fault-Tolerant Spaceborne Computer Compiler

GOVERNMENT AGENT: Space & Missile Systems Division/DY

DESCRIPTION: The AF Space and Missile Systems Organization (SAMSO) is developing a JOVIAL-73 compiler for use on the Fault-Tolerant Spaceborne Computer (FTSC) project. This compiler is currently under development via a contract by Rome Air Development Center, and is utilizing the DAIS JOVIAL-73 compiler as a baseline. The compiler will be a cross-compiler for the FTSC, and the compiler itself is written in JOVIAL-73.

MAJOR SOFTWARE FUNCTIONS:

- Compiler Code Generator
- Compiler Editor
- Compiler Optimizer

STATUS: This effort is currently underway.

SIZE OF EFFORT: Approximately 35K words of code will be required in addition to the existing DAIS compiler baseline.

COMPANIES INVOLVED: Computer Sciences Corporation, El Segundo CA

PROJECT: J-73 Language Evaluation

GOVERNMENT AGENT: AF Communications Service

DESCRIPTION: The Communication's Computer Programming Center (CCPC), Tinker AFB, is using the DAIS DEC-10 JOVIAL-73 compiler to perform an in-house evaluation of the JOVIAL-73 language. The objective is to assess the capabilities of J-73 for application to communications system problems. Small test cases are being implemented in J-73.

MAJOR SOFTWARE FUNCTION:

- Communications processing

STATUS: This effort is currently underway.

SIZE OF EFFORT: Unknown

COMPANIES INVOLVED: None

PROJECT: JOVIAL-73 Compiler Validation System

GOVERNMENT AGENT: AF Rome Air Development Center/IS

DESCRIPTION: The AF Rome Air Development Center (RADC), Griffiss AFB developed under contract a JOVIAL-73 Compiler Validator System (JCVS). The JCVS software was itself written in J-73, and performs testing of J-73 compilers by use of J-73 source programs which methodically exercise the complete language syntax/semantics.

MAJOR SOFTWARE FUNCTIONS:

- Language tests

STATUS: Complete

SIZE OF EFFORT: Approximately 62K words of code.

COMPANIES INVOLVED: Abacus Corporation, Santa Monica CA

SECTION III

JOVIAL-3B USAGE DATA

This section documents and summarizes the results of the survey of JOVIAL-3B language usage. Table (B)III-2 itemizes the results on the four projects that JOVIAL-3B has been used on to date. Also, each of these projects is covered in more detail in the subsequent pages. The author did not find any additional near term projects for which JOVIAL-3B is planned for usage. The salient points to note concerning the results are:

1. Approximately 143K words of JOVIAL-3B object code exists today, and an estimated 50K words more are currently under development.

2. Four companies (Boeing, Softech, General Dynamics, and Airborne Instrumentation Labs) have experience in development of JOVIAL-3B code.

3. Three major government organizations are involved in projects presently using JOVIAL-3B.

4. Four separate projects have, or are, using the JOVIAL-3B language.

5. The uses of JOVIAL-3B have been restricted to two applications areas, avionics and real-time executives. However, virtually all of this usage has been in the avionics area.

6. Only one company, Softech, has had any appreciable experience with building JOVIAL-3B compilers; however Boeing has had some experience with maintenance of the B-1 project compilers.

TABLE (B)III-2

CURRENT AND PLANNED JOVIAL-3B USERS

GOVERNMENT AGENCY	PROJECT	COMPANY	PROGRAM SIZE		STATUS
			DEVELOPED	PROPOSED	
ASD/YH	B-1 Offensive Software	Boeing	95K	-	Completed
	B-1 Defensive Software	Boeing & AIL	24K*	46K*	Under Development
ASD/YP	F-16 Fire Control Software	General Dynamics	20K	4K	Under Development
AFAL/AA	Operational Software Concept	General Dynamics & Softech	4K	-	Completed

*These figures are very approximate since limited data was available.

PROJECT: B-1 Offensive Software

GOVERNMENT AGENT: Air Force Aeronautical Systems Division/YH

DESCRIPTION: The Aeronautical Systems Division, Wright-Patterson AFB, is responsible for the development of the B-1 stragetic bomber aircraft. The Boeing Company has developed the primary Offensive Avionics for the B-1, and the two primary airborne computers are Singer SKC-2070's. Interconnection of these computers to the remainder of the avionics is primarily through a TDM data bus, similar to the MIL-STD-1553A scheme. Most of the Offensive Flight Software (75%) for the SKC-2070's is coded in JOVIAL-3B/O.

MAJOR SOFTWARE FUNCTIONS

- Controls/Displays
- Inertial Navigation
- Doppler Navigation
- Doppler-Inertial Navigation
- Error Control/Reconfiguration
- Weapon Delivery
- Target/Fix Taking
- Mode Control
- Integrated Test

STATUS: This effort is completed.

SIZE OF EFFORT: The final J-3B software is approximately 95K words of object code.

COMPANIES INVOLVED: Boeing, Seattle, Washington, developed the airborne software, and Softech, Waltham, Massachusetts, developed the JOVIAL-3B compiler.

PROJECT: B-1 Defensive Software

GOVERNMENT AGENT: Air Force Aeronautical Systems Division/YH

DESCRIPTION: The Aeronautical Systems Division, Wright-Patterson AFB, is responsible for the development of the B-1 stragetic bomber. The Boeing Company is the integrator for Defensive Avionics on the B-1 using primarily the Litton LC-4516D computer and an additional Singer SKC-2070. AIL is the vendor for the defensive subsystem equipments and software. Interconnection of these computers is via the same TDM data bus that is used for the offensive avionics.

MAJOR SOFTWARE FUNCTIONS

- Controls/Displays
- Data Base Management
- Emitter Recognition
- Emitter Identification
- Jammer Management
- Power Management

STATUS: This effort is presently underway. Approximately one-third of the software is completed.

SIZE OF EFFORT: The final software size is projected to be approximately 30K words for the SKC-2070 and 40K words for the LC-4516D.

COMPANIES INVOLVED: Boeing, Seattle, Washington, is the integrator, and AIL, Long Island, New York, is the subsystem vendor.

PROJECT: F-16 Fire Control

GOVERNMENT AGENT: Air Force Aeronautical Systems Division/YP

DESCRIPTION: The Aeronautical Systems Division, Wright-Patterson AFB, is responsible for the development of the F-16 air superiority fighter. One of the computers onboard, the fire control computer, will have a major portion of the functions programmed in JOVIAL-3B. The computer, a DELCO Magic-362F, is interconnected with other aircraft avionics primarily through a MIL-STD-1553A data bus.

MAJOR SOFTWARE FUNCTIONS:

- Weapon Delivery
- Navigation Fix-taking
- Stores Management
- HUD Control
- Energy Management

STATUS: This effort is presently underway, and major portions of the flight software have been completed.

SIZE OF EFFORT: The final J-3B software will be approximately 24K words. Of this an estimated 20K words now exist.

COMPANIES INVOLVED: General Dynamics, Fort Worth, Texas, is developing the airborne software, and Softech, Waltham Massachusetts, developed the compiler.

PROJECT: Operational Software Concept

GOVERNMENT AGENT: Air Force Avionics Laboratory/AA

DESCRIPTION: The Operational Software Concept (OSC) is an advanced development effort in the Air Force Avionics Lab, Wright-Patterson AFB. This is a multi-phased effort to develop and demonstrate avionic flight software design and implementation methodologies which will help reduce the life-cycle-cost of flight software. This project consists of both contractual and in-house efforts. Early OSC efforts resulted in a prototype flight executive developed in JOVIAL-3B.

FUNCTIONS:

- Executive Software

STATUS: Completed

SOFTWARE SIZE: Approximately 4K words of object code produced from J-3B source.

COMPANIES INVOLVED: General Dynamics, Fort Worth, Texas, and Softech, Waltham, Massachusetts

SECTION IV

JOVIAL-73 COMPILERS

This section, and the subsequent section on JOVIAL-3B compilers, documents the results of a survey which was made to assess the current status of JOVIAL-73 and JOVIAL-3B compilers. For each language there are a number of compilers and cross-compilers which either currently exist or are presently under development. This section will delineate and discuss the JOVIAL-73 compilers.

In order to be specific in naming these compilers, a convention was developed and used throughout the following sections. For example, for a JOVIAL-73 compiler the naming convention is:

"JOVIAL-73 (host, target)"

where "host" is the host computer on which the compiler executes, and "target" is the target computer for which the compiler produces object code. In the instance where the host computer differs from the target computer, the terminology "cross-compiler" is used. Cross-compliers have become the normal approach to developing compilers for flight computers due to the limited size and limited generality of most flight computer configurations.

Also some problems arose in "counting" the number of compilers for each language, and certain ground rules had to be made about what constitutes a "different" compiler. For instance, the JOVIAL-3B compiler on the IBM-360 has evolved over the years to support three dialects of this language; J-3B/0, J-3B/1, and J-3B/2. However, since these three dialects are upward compatible and the same basic compiler applies to all, this was counted as only one compiler. A similar situation exists for the JOVIAL-73/S compiler on the DEC-10 which is a "sub-compiler" of the full JOVIAL-73/I DEC-10 compiler. Also the JOVIAL-73/I compiler on the DEC-10

runs under two operating systems, the DEC-10 operating system and the DAIS Support Software System. However, the same basic compiler supports both operating system capabilities and thus was counted only once.

Also, some attempt has been made to distinguish between the "quality" of these compilers by indicating whether the particular compiler is a "production" compiler or not. The term "production" means that the compiler was designed and developed to be used on a daily basis as a production tool. This implies that the compiler is relatively sophisticated, is a highly reliable tool, produces optimized object code, and makes good utilization of the host machine resources. To the contrary, compilers not termed as "production" tools can be considered as either prototypes or interim compilers.

Table (B)IV-1 itemizes the JOVIAL-73 compilers that either exist today or that are currently under development. Three separate subsets of JOVIAL-73 have been implemented: (1) JOVIAL-73/M which is a near subset of JOVIAL-73/I; (2) JOVIAL-73/S which is a proper subset of JOVIAL-73/I; and (3) JOVIAL-73/I which is a proper subset of the full JOVIAL-73 language. These compilers are presently hosted on three major host machines and produce code for five target computers. Two other compilers are under development.

Figure (B)IV-1 presents somewhat of a different view of these compilers, more from a lineage viewpoint. As can be easily seen there are three separate "root" compilers, with the majority of the compilers evolving from the DAIS Project developmental efforts (top line).

In summary, there are currently five JOVIAL-73 compilers in existence today, implementing varying subsets of the full JOVIAL-73 language. These compilers are on three major host computers and generate code for five different target computers. There are two cross compilers currently

TABLE (B)IV-1: JOVIAL-73 COMPILERS

HOST MACHINE	TARGET MACHINE: UNIVAC 1108	DEC-10	IBM 360/370	WESTINGHOUSE AN/AYK-15	AIR-RESEARCH RF-20	RAYTHEON FTSC	WESTINGHOUSE EAR COMPUTER
UNIVAC 1108	J-73/S	J-73/S					
DEC-10		J-73/I & S		J-73/I		J-73/I*	
IBM 360/370					J-73/M		
UNIVAC 1110							J-73/I*

*Compilers currently under development

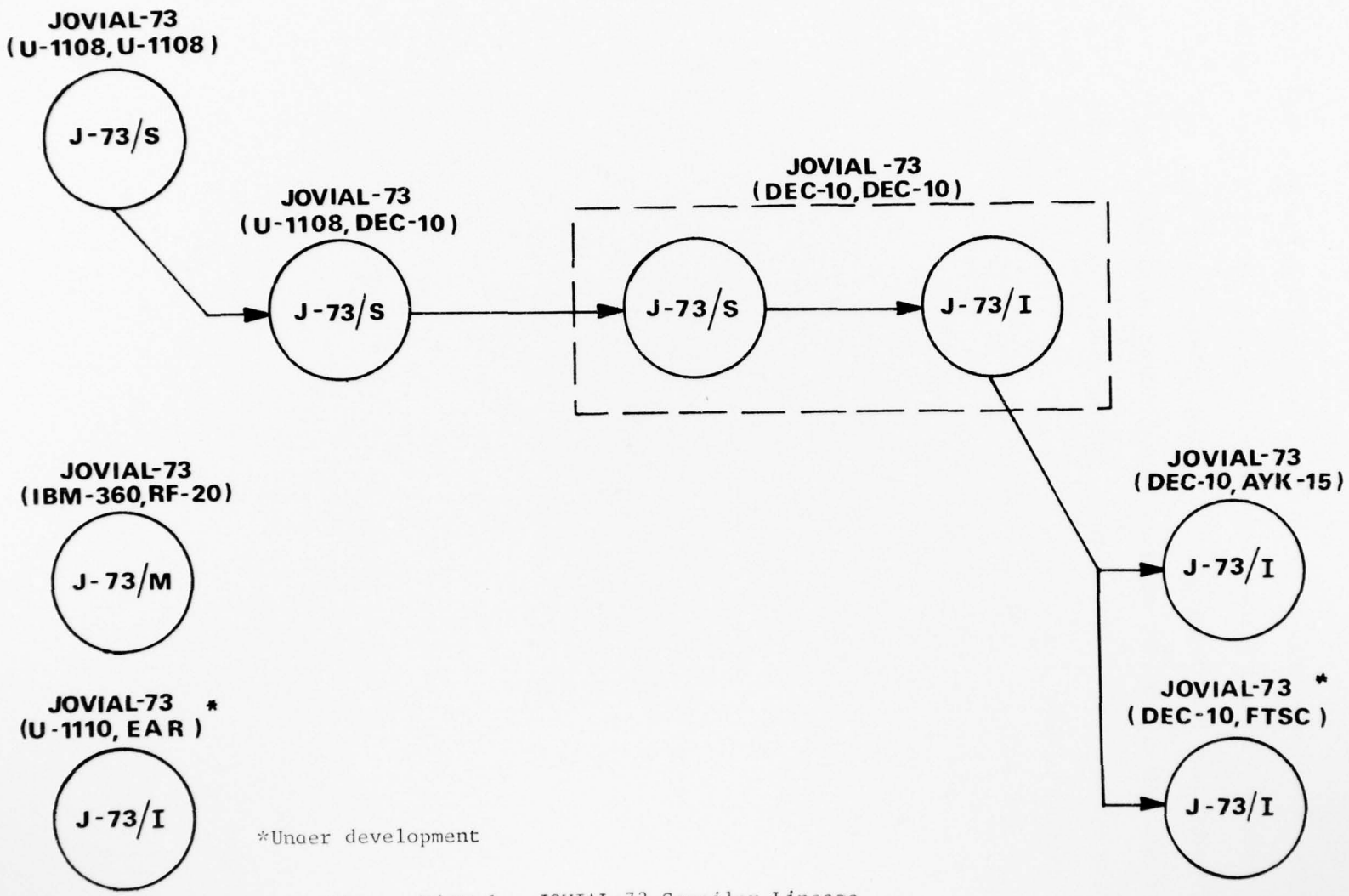


Figure (B)IV-1. JOVIAL-73 Compiler Lineage

under development, one to be on another major host computer. Also, three companies (Computer Sciences Corp., Westinghouse, and Proprietary Software Systems) have been involved in supplying JOVIAL-73 compilers, and two others (TRW and ABACUS) are currently involved in JOVIAL-73 compiler maintenance. The following seven sections discuss each of these compilers in more detail.

COMPILER: JOVIAL-73 (U-1108, U-1108) Compiler

Language: JOVIAL-73/S

Description: This compiler was developed on the UNIVAL-1108 computer system as an outgrowth of some early JOVIAL-73 development work sponsored partially by Rome Air Development Center, Griffiss AFB. The compiler was subsequently used to support early JOVIAL-73/I compiler development activities for the DAIS Project. The compiler implements the JOVIAL-73/S language, which is a proper subset of JOVIAL-73/I. The compiler itself is coded in the JOVIAL-73/S language. This compiler is not a "production" class compiler.

Vendor: Computer Sciences Corporation, El Segundo, California.

COMPILER: JOVIAL-73 (U-1108, DEC-10) Cross-Compiler

Language: JOVIAL-73/S

Description: This cross-compiler was developed from the JOVIAL-73 (U-1108, U-1108) Compiler as part of an interim measure for supplying the final J-73 compiler for the Digital Avionic Information System (DAIS) Project, A.F. Avionics Lab, Wright-Patterson AFB. This compiler implemented the JOVIAL-73/S language which is a proper subset of JOVIAL-73/I. Since this cross-compiler was an interim measure only, it was not developed as a "production" compiler. The compiler itself is written in JOVIAL-73/S.

Vendor: Computer Sciences Corporation, El Segundo, California

Compiler: JOVIAL-73 (DEC-10, DEC-10) Compiler

Languages: JOVIAL-73/S

JOVIAL-73/I

Description: This compiler implements the JOVIAL-73/I language and was produced for the Digital Avionic Information System (DAIS) Project, A.F. Avionics Lab, Wright-Patterson AFB. The purpose was to provide a compiler and language that would support the development of support software needed for the DAIS project, and also to provide for algorithmetic checkout of DAIS flight software. The compiler is itself written in JOVIAL-73/I. The compiler was designed and developed to be used in a "production" environment, and provides extensive optimization for producing highly efficient object code. The compiler and all associated software is fully government owned and documented. Two versions of the compiler exist for operation under different operating system environments, and versions have also been produced to implement both the JOVIAL-73/S and JOVIAL-73/I Language subsets.

Vendor: *Computer Sciences Corporation, El Segundo, California

*Maintenance is currently being performed by TRW with a subcontract to ABACUS.

COMPILER: JOVIAL-73 (DEC-10, AN/AYK-15) Cross-Compiler

Language: JOVIAL-73/I

Description: This cross-compiler was produced for the Digital Avionics Information System (DAIS) Project, A.F. Avionics Lab, Wright-Patterson AFB. The purpose was to provide a cross-compilation capability (on the DEC-10) for producing DAIS flight software for the AN/AYK-15 airborne computer. The cross-compiler is written in JOVIAL-73/I, and is an extension of the previously developed JOVIAL-73 (DEC-10, DEC-10) compiler. This is also classed as a "production" compiler and uses extensive optimization to generate highly efficient object code. The cross-compiler and all associated software is fully government owned and documented. Versions of the compiler exist for execution under different operating system environments.

Vendor: *Computer Sciences Corporation, El Segundo, California

*Maintenance is currently being performed by TRW with a subcontract to ABACUS.

COMPILER: JOVIAL-73 (DEC-10, FTSC) Cross-Compiler

Language: JOVIAL-73/I

Description: This cross-compiler is currently under development for the Air Force Space and Missile Systems Organization (SAMSO), Los Angeles, California. The compiler is being developed under contract to Rome Air Development Center, Griffiss AFB, and will provide a "production" oriented cross-compiler for the Raytheon Fault-Tolerant Spaceborne Computer (FTSC). The compiler is a derivative of the JOVIAL-73 (DEC-10, DEC-10) compiler produced for DAIS, and implements the JOVIAL-73/I language. The compiler itself is written in JOVIAL-73/I and fully documented and owned by the Air Force.

Vendor: Computer Sciences Corporation, El Segundo, California

COMPILER: JOVIAL-73 (IBM-360, RF-20) Cross-Compiler

Language: JOVIAL-73/M

Description: This cross-compiler was developed as part of a prototype development effort for the Advanced Stragetic Air Launch Missile (ASALM) Project. Aeronautical Systems Division, Wright-Patterson AFB. The purpose was to provide a very simple, easily implemented, cross-compiler for the Garett AirResearch RF-20 flight computer being used on the prototype ASALM. The host computer was an IBM-360, and the language, JOVIAL-73/M, is a near-subset of JOVIAL-73/I. The compiler is not of "production" quality and no emphasis was placed on producing efficient code. The implementation is based on proprietary software products and is implemented in the TILT language. The compiler was developed for Martin-Marietta Orlando, Florida; the ASALM prime contractor.

Vendor: Proprietary Software Systems, Inc., Beverly Hills, California.

COMPILER: JOVIAL-73 (U-1110, EAR) Cross-Compiler

Language: JOVIAL-73/I

Description: This cross-compiler is being developed by the Electronically Agile Radar (EAR) Project Office, A.F. Avionics Lab, Wright-Patterson AFB. The compiler is currently under development and will implement the JOVIAL-73/I language. It is being written in FORTRAN, being implemented on the UNIVAC-1110, and is intended to be a "production" compiler.

Vendor: Westinghouse Corporation, Baltimore, Maryland.

SECTION V

JOVIAL-3B COMPILERS

This section documents the results of a survey which was made to assess the current status of JOVIAL-3B compilers. Table (B)V-1 itemizes the various compilers by host and target computers, and also indicates the various JOVIAL-3B dialects implemented. The JOVIAL-3B/0 dialect was the original JOVIAL-3B language, but it has since been twice expanded to two larger superset languages; JOVIAL-3B/1 and JOVIAL-3B/2.

Figure (B)V-1 shows the lineage of JOVIAL-3B compilers. The JOVIAL-3B (IBM-360, IBM-360) compiler has been the "root" compiler for all JOVIAL-3B work to date, and all three dialects are processed by this compiler. As can be seen, several cross-compilers have been derived from this basic compiler.

In summary, there are currently five different JOVIAL-3B compilers, and all are hosted on the IBM-360/370 series computers. To date, only Softech has been involved in supplying JOVIAL-3B compilers, but Boeing has been involved in some JOVIAL-3B compiler maintenance activities. The following five sections discuss each of these compilers in more detail.

TABLE (B)V-1

JOVIAL-3B COMPILER

HOST MACHINE	TARGET MACHINE: IBM 360/370	SINGER SKC-2070	IBM 4PI-CP2	LITTON LC-4516D	DELCO M362F	
IBM 360/370	J-3B/0,1&2	J-3B/0&1	J-3B/0	J-3B/1	J-3B/2	

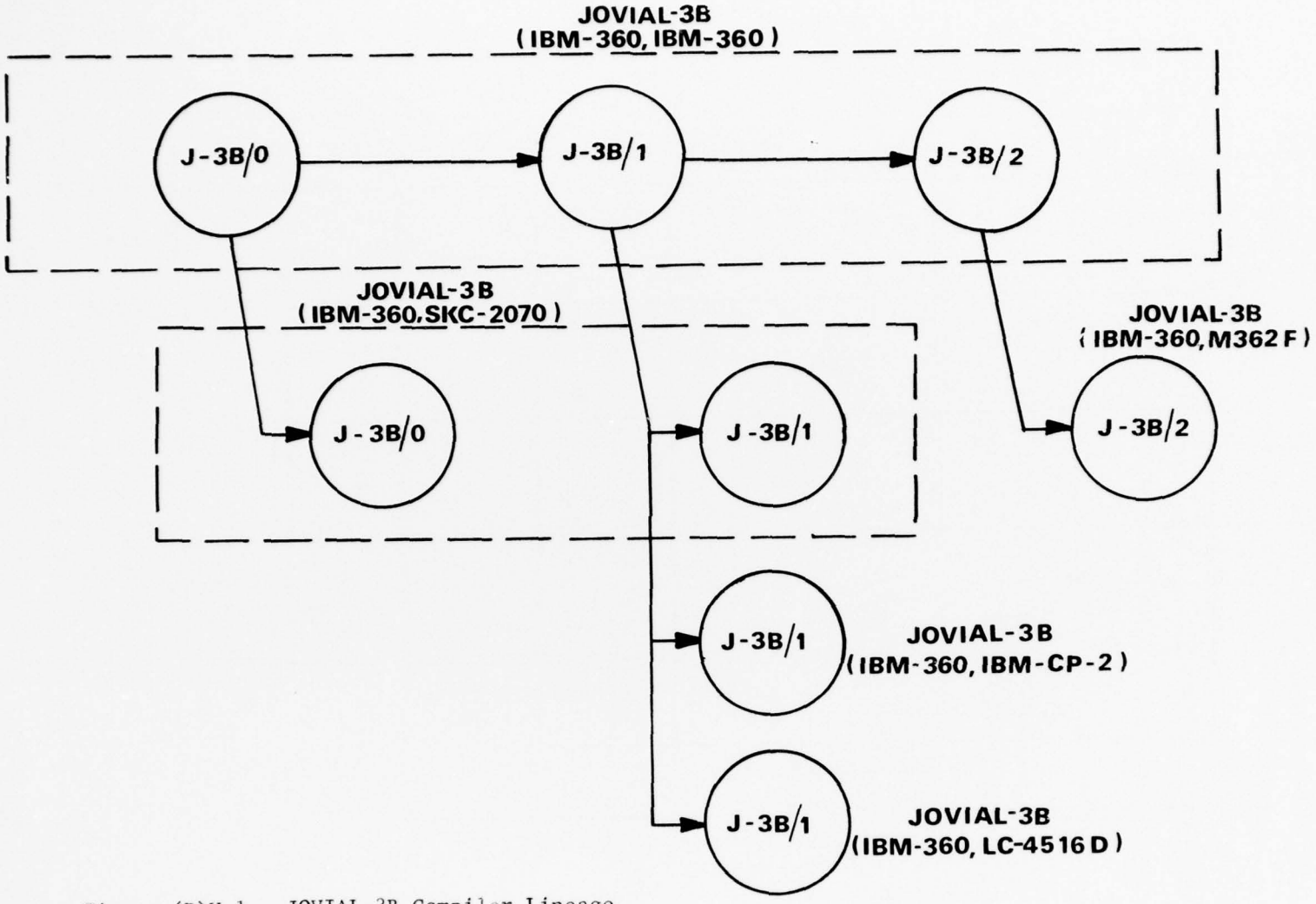


Figure (B)V-1. JOVIAL-3B Compiler Lineage

COMPILER: JOVIAL-3B (IBM-360, IBM-360) Compiler

LANGUAGES: JOVIAL-3B/0

JOVIAL-3B/1

JOVIAL-3B/2

DESCRIPTION: This JOVIAL-3B compiler is hosted on, and produces code for, the IBM-360/370 series computers. It was originally developed for the B-1 System Program Office, Aeronautical Systems Division, Wright-Patterson AFB. The IBM-360 version was originally used for B-1 flight software algorithm development, and subsequently the flight software was cross-compiled for the B-1 flight computer (Singer SKC-2070). Several versions of this compiler have been developed over the past few years and three dialects of the J-3B language are implemented (J-3B/0, J-3B/1, and J-3B/2). Also, interfaces exist for different operating system environments. The compiler is implemented in the AED language and is a "production" class compiler. Full documentation exists on the compiler implementation, but it does depend on the AED compiler which is only available from Softech.

VENDOR: *Softech, Waltham, Massachusetts

*Some compiler maintenance activities have also been performed by Boeing.

COMPILER: JOVIAL-3B (IBM-360, SKC-2070) Cross-Compiler

Languages: JOVIAL-3B/0

JOVIAL-3B/1

Description: The JOVIAL-3B, SKC-2070 cross-compiler was developed for the B-1 System Program Office, Aeronautical Systems Dvision, Wright-Patterson AFB. The compiler implements two J-3B dialects (J-3B/0 and most of the J-3B/1) and generates code for the B-1 flight computer (Singer SKC-2070). The cross-complier is a derivative of the JOVIAL-3B (IBM-360, IBM-360) compiler and is thus written in the same AED language. This is a "production" class compiler.

Vendor: *Softech, Waltham, Massachusetts

*Some compiler maintenance activities have also been performed by Boeing.

COMPILER: JOVIAL-3B (IBM-360, IBM-4PI) Cross-Compiler

Language: JOVIAL-3B/0

Description: The JOVIAL-3B cross-compiler for the IBM-4PI/CP-2 airborne computer was developed for the A.F. Avionics Lab, Wright-Patterson AFB. This compiler implemented the JOVIAL-3B/0 dialect, and was developed as a derivative **of** the previous JOVIAL-3B(IBM-360, IBM-360) compiler. The compiler was **used** during initial phases of the Operational Software Concepts (OSC) Project which demonstrated multiple avionic computer concepts using IBM-4PI/CP-2 computers. The compiler is also written in the AED language.

Vendor: Softech, Waltham, Massachusetts

COMPILER: JOVIAL-3B (IBM-360, LC-4516D) Cross-Compiler

Language: JOVIAL-3B/1

Description: The JOVIAL-3B cross-compiler for the Litton LC-4516D airborne computer was developed for the B-1 System Program Office, Aeronautical Systems Division, Wright-Patterson AFB. This compiler implements the JOVIAL-3B/1 dialect, and was developed as a derivative of the previous JOVIAL-3B (IBM-360, IBM-360) compiler. The compiler is currently being used to support coding of B-1 Defensive System Software which uses LC-4516D flight computers. The compiler is written in the AED language.

Vendor: Softech, Waltham, Massachusetts

COMPILER: JOVIAL-3B (IBM-360, M362F) Cross-Compiler

Language: JOVIAL-3B/2

Description: The JOVIAL-3B cross-compiler for the DELCO-M362F airborne computer was developed for the F-16 System Program Office, Aeronautical Systems Division, Wright-Patterson AFB. This compiler implements the JOVIAL-3B/2 dialect, and was developed as a derivative of the previous JOVIAL-3B(IBM-360, IBM-360) compiler. The compiler is currently being used to support flight software development of the F-16 fire control computer (DELCO Magic 362F). The compiler is written in the AED language.

Vendor: Softech, Waltham, Massachusetts

SECTION VI

SUMMARY AND CONCLUSIONS

This section summarizes the results of the data that were itemized in the previous sections, and also documents a few conclusions about the relative status of JOVIAL-73 and JOVIAL-3B. Table(B)VI-1 lists nine parameters that were selected as a basis for comparison; obviously, not all parameters are of equal importance. For all but one (number of "production" class compilers to date), the JOVIAL-73 language data equals or exceeds that of JOVIAL-3B! A few of the more important aspects are discussed below:

1. Language "Maturity": The parameters associated with how extensively and for what applications the language has been used are a good measure of the "maturity" of the language. The particular parameters of 1, 5, 8, 9, 10, and 11 (Table(B)VI-1) are indicators of such maturity. As is easily seen the JOVIAL-73 language can be easily ranked as a more mature language than JOVIAL-3B.

2. User Base: An important aspect in Air Force standardization attempts, the size of the "user base" for a particular language is an important issue. Parameters such as 2, 3, 4, 6, and 7 (Table(B)VI-1) are all indicators of the size of this user base. Again, in terms of these measures JOVIAL-73 is a much more widely used, experienced, and accepted language.

TABLE (B)VI-1

SUMMARY COMPARISON DATA

PARAMETER		METRIC	LANGUAGE			
			JOVIAL-73		JOVIAL-3B	
NUMBER	Description		To Date	Additional Near Term	To Date	Additional Near Term
1	Language usage extent	Words of object code produced from JOVIAL source	569K*	399K	143K	50K
2	Contractors involved in language usage	Number of companies	7	2	4	0
3	Major government agencies involved in language usage	Number of agencies	5	3	3	0
4	Projects using language	Number of projects	8	6	4	0
5	Various applications for which language used	Number of different application areas	4	1	2	0
6	Experienced compiler vendors for language	Number of companies	3	0	1	0
7	Experienced compiler "Maintainers" for language	Number of companies	5	0	2	0
8	"Production" class compilers	Number of compilers	2	2	5	0
9	Total compilers	Number of compilers	5	2	5	0
10	Different Hosts for "Production" compilers	Number of Host computers	1	1	1	0
11	Different Hosts for all compilers	Number of Host computers	3	1	1	0

* Largely Avionics Support Software

APPENDIX C

JOVIAL-73 R&D EFFORTS

There are currently several efforts for JOVIAL-73 at RADC which are either ongoing or planned to start in the near future. These efforts are all contractual except one (JCVS), and are being performed out of the Information Sciences Division (RADC/IS). The following six sections itemize these major JOVIAL-73 efforts. This information was compiled from RADC TPO documents, DDR&E Apportionment Review material, and conversations with RADC/IS personnel.

1. JOVIAL Compiler Validation System (JCVS): The JCVS is a JOVIAL-73 compiler test tool that provides for a fairly rigorous validation of a JOVIAL-73 compiler in the language processing area. JCVS is itself a collection of JOVIAL-73 programs that methodically exercise each and every construct in the JOVIAL-73 language. These JCVS tests are then compilable programs that are in turn executed, and the results are an analysis of how well the JCVS code executed. This type of test not only provides for exercising a compiler to find faults, but it is also an aid in insuring that various JOVIAL-73 compilers will produce the same results. This is a very important tool for an effective language standardization program. JCVS is currently developed, but in-house efforts are underway at RADC to further enhance the JCVS programs.

2. JOVIAL Automated Validation System (JAVS): JAVS was originally developed to be used as a JOVIAL-3 software validation aid. It functions as a pre-processor and instruments JOVIAL-3 source code prior to a compilation. JAVS pre-processing inserts additional JOVIAL-3 code into the JOVIAL-3 source file which is under test, and this inserted JAVS code collects run-time data on the execution of the software. In turn, the data collected by JAVS is analyzed and indicates the control paths executed by the software, the data values that existed at control decision points, and indications of test data values needed to better exercise the software. JAVS is currently implemented for JOVIAL-3 programs, but RADC has proposed to convert JCVS to use with JOVIAL-73.

This effort is planned for FY78 and is estimated to be approximately $100K.

3. JOVIAL-73/I Tutorial Programming Manual: At the request of AFAL, RADC has an effort underway to produce a tutorial Programming Manual for JOVIAL-73/I. This manual is to be used as a teaching aid in learning the JOVIAL-73/I language. Currently available manuals on JOVIAL-73 are classed as "reference manuals" and can only be effectively used by experienced programmers. This effort is funded for approximately $25K and scheduled for completion by December 1976.

4. JOVIAL-73 Compiler Implementation Tool (JOCIT-73): JOCIT-73 is a JOVIAL-73 compiler development project to produce compiler generation tools for easily and quickly building JOVIAL-73 compilers. A similar RADC effort was completed in 1973 for the JOVIAL-3 language. Initial use of the JOCIT-73 is planned to be for the RPV Project Office for use on the RPV Mission Control System (RMCS). The effort is planned to begin in FY77 with an expenditure of $550K and continue through FY78 with $750K.

5. JOVIAL-73/I Compiler: At the request of SAMSO, RADC is presently procuring a JOVIAL-73/I cross-compiler for the Fault Tolerant Spaceborne Computer (FTSC). The compiler is being developed from the DAIS JOVIAL-73/I compiler and will be resident on the DEC-10 computer system. The effort is approximately $100K.

6. Language Control Facility: Plans are currently being made at RADC to initiate a language control facility effort to address both JOVIAL-3 and JOVIAL-73. The exact functions of this facility are not defined at this time, and the amount of the effort focused at JOVIAL-73 versus JOVIAL-3 is not known. The minimum effort on JOVIAL-73 would cover configuration control of the language specification. This is proposed for 6.4 type funding at $200K in FY77 and $300K in FY78.

☆ U. S. GOVERNMENT PRINTING OFFICE: 1977 — 757-001/292